**Aim**

Investigate chaos in Lorenz equation using computational physics simulation and techniques using python.

**Summary**

The Lorenz Equations (above) provide a simple and yet significant area of study as it provides scientists a chaotic dynamical outlook of a system. We will be investigating different parameters of the Lorenz Equation to produce some of the most fascinating graphical outcomes of the trajectory of this 3 dimensional ODE. We will be applying what we learned in class using Runge-Kutta method to solve the non-analytical equations arising from its non-linearity.

This report will revolve around the book created by Steven H. Strogatz titled “Nonlinear Dynamics and chaos” with the help of computational knowledge and simulations in Python obtained from Computation Physics.

**Introduction**

First derived by Edward Lorenz in 1963, the three-dimensional system came from a simplified model of convection system of the atmosphere. The trajectories of the system also converge into stable cycles which are now called strange attractors relating to fractals. These simple looking equations could have extreme dynamics over a wide range of parameters and producing chaos.

Where , r, b are parameters that we will be playing with.

is known as the Prandtl number.

r is known as Rayleigh number, dimensionless measure of difference in temperature of the top and bottom of the fluid.

b has no name, but also an important parameter. In convection problems, this is related to the aspect ratio of the rolls (Strogatz, 1994).

Popularized by media, even non mathematical or physics people are enchanted by the beauty of the chaos in the Lorenz equation. These fascination might attribute to the appeal of the patterns they produce and hence inspiring people with creative ideas and philosophy. In this report, we will be investigating what causes these patterns by exploring multiple parameters of the Lorenz Equations and tuning gathering data for each change in the system. Furthermore, we will look into how these patterns are produced and what they mean.

Chaos theory is defined as the qualitative study of unstable aperiodic behaviour in deterministic nonlinear dynamical systems (Kellert, 1993).

**Survey**

In forming the code, it was initially tough to get the simulations running. In order to have a neater way of the simulation, I decided to gather these function into a class, to call the data as self.res[i] where i = 0,1,2 for convenience in playing with the coordinates obtained.

In addition, I have also made an interactive python GUI app that allows user to take in different values of r, sigma and b using the module Tkinter. The 3D graph is embedded into the Tkinter frame so that the user can manually enter parameters of their choice.

A screenshot of a cell phone

Description automatically generated

Fig. 1. Main UI of Lorenz Graph Plotter.

To simplify the program and improve running time of the program, we are only simulating **one** particle undergoing the Lorenz trajectory.

A screenshot of a cell phone

Description automatically generated

One problem that arised is that, the canvas used to draw the data of the graph does not automatically clear when it is pressed “Run”. The method ﻿“FigureCanvasTkAgg” does not contain any “clear”, or “delete” attributes in the built-in method. Hence the simplest way to clear the data is to press “Clear” button on the app.

**Code Organization**

The basic structure of the code is fairly simple, I have called the initial values to 0.

Self-recurring functions to estimate the trajectory of the ODEs and outputting it into 3-Dimensional graph using matplotlib.

Solving the ODE require the execution of the Runge-Kutta to the fourth order method to serve as a good estimate approximation of the trajectory in phase space.

**Results**

To begin showing the results, let us breakdown the Lorenz equation into its two different fixed points. This is simply obtained by setting . Solving these equations will yield two fixed points that Lorenz called C+ and C-.

C+ ~ (

C- ~ (

**Discussion**

Using the formula below to compute the next value from the previous value . We introduce a step size *h* and where n are positive integers.

This method of estimation has a local truncational error on the order of while the total accumulated error is on the order .

This method is known as the Runge Kutta in the fourth order.

**References**

Strogatz, H. S. (1994). Nonlinear Dynamics and chaos.

Kellert, H. S. (1993). In the Wake of Chaos: Unpredictable Order in Dynamical Systems. p.2

**Appendix**

(3 marks) a one-paragraph summary of the class project;

(3 marks) surveys of the problems and methods;

(4 marks) a big picture on the organization of your code;

(10 marks) final versions of your code in the Appendix;

(4 marks) report your final results;

(3 marks) discuss the physics of these results; and

(3 marks) list your references.